Распознавание образов

Рекомендуется выполнять работы в программе Jupyter Notebook. Скачать ее можно по ссылке https://www.anaconda.com/distribution/ . Инструкция по скачиванию (для ОС Windows): https://www.coursera.org/lecture/mathematics-and-python/kak-ustanovit-anakondu-windows-KN0Sf .

Для выполнения работы будет использоваться функционал библиотеки Python scikit-learn. Ссылка на документацию: https://scikit-learn.org/stable/documentation.html . Эта библиотека содержит встроенные датасеты, нацеленные для работы с регрессией и классификацией.

Лабораторная работа № 1

Классификация данных

Теоретические сведения

Импорт необходимых библиотек. Однозначно будут использоваться следующие импорты:

```
from sklearn import datasets
from sklearn import tree
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
```

Загрузка датасета:

```
iris = datasets.load_iris()
```

Датасет имеет:

- названия признаков (*iris.feature_names*);
- значения признаков (*iris.data*);
- категория, к которой принадлежит объект с данными признаками (*iris.target*);
- название категории, к которой принадлежит объект с данными признаками (*iris.target_names*).

Формирование обучающей и тестовой выборки. Необходимо сформировать массивы:

- значений признаков обучающей выборки;

- категорий обучающей выборки;
- значений признаков тестовой выборки;
- категорий тестовой выборки.

Построение дерева решений и классификатора:

clf = tree.DecisionTreeClassifier()

clf = clf.fit(<массив значений признаков обучающей выборки>, <массив категорий обучающей выборки>)

Классификация данных:

clf.predict([<признаки объекта>])

Расчет параметров для оценки классификации:

accuracy_score(<массив реальных категорий>, <массив предсказанных категорий>)

precision_score(<массив реальных категорий>, <массив предсказанных категорий>)

recall_score(<массив реальных категорий>, <массив предсказанных категорий>)

f1_score(<массив реальных категорий>, <массив предсказанных категорий>)

Ход работы:

1. разбить заданный датасет на тренировочную и тестовую выборку в соотношении 70:30 случайным образом; данные выборки не должны пересекаться;
2. на основе обучающей выборки построить дерево решений;
3. на основе дерева решения построить классификатор;
4. проверить работу классификатора на обучающей и тестовой выборках;
5. рассчитать и оценить *accuracy*, *precision*, *recall*, *F-measure* распознавания на обеих выборках.

Лабораторная работа № 2

Кластеризация данных

Теоретические сведения

Импорт необходимых библиотек. Однозначно будут использоваться следующие импорты:

from sklearn import datasets

from sklearn.cluster import KMeans

from sklearn.metrics import adjusted_rand_score

Загрузка датасета:

iris = datasets.load_iris()

Визуализация выбранных признаков датасета и принадлежности объектов к определенным категориям согласно значениям этих признаков:

X = iris.data[:, <индексы выбранных признаков>]

y = iris.target

fig, axes = plt.subplots(1, 1, figsize=(8,4))

axes.scatter(X[:,0], X[:,1], c=y)

axes.set_xlabel("<название первого выбранного признака>", fontsize=14)

axes.set_ylabel("<название второго выбранного признака>", fontsize=14)

Кластеризация методом *k*-средних:

kmeans = KMeans(n_clusters = <число кластеров>)

km.fit(X)

*labels = kmeans.labels_*

Расчет индекса Рэнда:

adjusted_rand_score(<реальные данные>, <результаты кластеризации>)

Ход работы:

1. выбрать любые 2 признака, на основе которых будет проводиться визуализация;
2. визуализировать данные признаки из датасета и принадлежность объектов к определенным категориям согласно значениям этих признаков;
3. произвести кластеризацию выбранных признаков методом *k*-средних;
4. визуализировать результаты кластеризации;
5. рассчитать и оценить индекс Рэнда.

Структура отчетов:

1. титульный лист;
2. цель работы;
3. дерево решений;
4. программный код с комментариями;

5. вывод.

Работа с датасетом осуществляется в соответствии с вариантом. Вычисление варианта: *i % 3*, где *i* – порядковый номер студента в ИСУ.

Варианты:

- 0 – load_iris;
- 1 – load_wine;
- 2 – load_breast_cancer.

К защите лабораторной работы необходимо знать:

- описание датасета, по которому выполнялась работа (размер, признаки, категории);
- предназначение использованных функций и аргументов в программном коде;
- лекционный материал.

Перед защитой отчеты отправлять на почту yulia1344@gmail.com. В теме письма указывать: «Распознавание образов. Лабораторная работа <номер>. <Номер группы> <ФИО>».